AFRL-RI-RS-TR-2017-235

INTERNET GEO-LOCATION

DUKE UNIVERSITY

DECEMBER 2017

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2017-235 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
ROBERT KAMINSKI
Work Unit Manager

/ S /
WARREN H. DEBANY, JR
Technical Advisor, Information
Exploitation and Operations Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
DEC 2017	FINAL TECHNICAL REPORT	MAY 2014 – MAY 2017

4. TITLE AND SUBTITLE

INTERNET GEO-LOCATION

5a. CONTRACT NUMBER FA8750-14-2-0150

5b. GRANT NUMBER N/A

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Bruce Maggs

5d. PROJECT NUMBER NSA4

5e. TASK NUMBER DU

5f. WORK UNIT NUMBER KE

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Duke University
Office of Research Administration
2200 W. Main St. STE 710
Durham NC 27705-4677

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIG
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER AFRL-RI-RS-TR-2017-235

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The geolocation system Alidade, developed with support from this research contract, is a passive IP geolocation system that is fundamentally different from all prior research systems since it computes predictions for the entire IP address space while absolutely refraining from issuing any measurement probes of its own, either before or after it is presented with the IP addresses. Although the system outputs (for each IP address) a point-based prediction for comparing its predictions with that of the other geolocation systems, Alidade's geolocation prediction is a polygonal feasible region that represents all possible locations of the IP address.

15. SUBJECT TERMS

IP address space, IP geolocation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					ROBERT KAMINSKI
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	UU	37	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Table of Contents

List of Figures

List of Tables

1 Summary

IP geolocation is defined as the process of determining the geographic location of an IP address based on measurement and/or non-measurement data. A geolocation system, for instance, may predict that the IP address 152.3.137.235 is located in Durham, NC and output the latitude-longitude coordinate 35.96, −78.94 associated with this location. Although the IP geolocation problem has been studied extensively [2,10,18,19,28,29], the systems discussed in academic literature have relied, typically, on *active probing* where the system issues measurement probes to gather delay-based measurements required for making the geolocation prediction. The geolocation system *Alidade*, developed with support from this research contract, is a *passive* IP geolocation system that is fundamentally different from all prior research systems since it computes predictions for the entire IP address space while absolutely refraining from issuing any measurement probes of its own, either before or after it is presented with the IP addresses. Although the system outputs (for each IP address) a point-based prediction for comparing its predictions with that of the other geolocation systems, Alidade's geolocation prediction is a polygonal *feasible* region that represents all possible locations of the IP address. Figure 1 shows an example of a geolocation prediction made by Alidade.

Some of the contributions we made during the period of this contract, are as follows.

- We provide deep insights into the IP geolocation techniques used in making geolocation predictions. By exposing the internals of the system, we aim to help users to make an informed choice on the use of the geolocation predictions. This design choice also potentially enables use of machine-learning techniques to automatically explore the space of heuristics and determine better ways of generating geolocation predictions.
- We make extensive use of city-level and state-level location hints. Using high-resolution shape files that accurately represent the different states, particular in the United States, can help in reducing the size of the feasible region of the prediction and increasing the accuracy of point-based predictions.
- We "seed" the system using a hand-cultivated set of ground-truth location data obtained from an industrial source and measure its impact on geolocation accuracy. While it is common-knowledge that geolocation systems have numerous "hard-coded" IP-address-to-location mappings, it is not clear to what extent such answers help and where this technique fails. Answers to such questions can help, for instance, in geolocating the IPv6 address space.
- We offer a simple interface to lookup geolocation predictions and inspect in detail the algorithm use for generating the predictions.

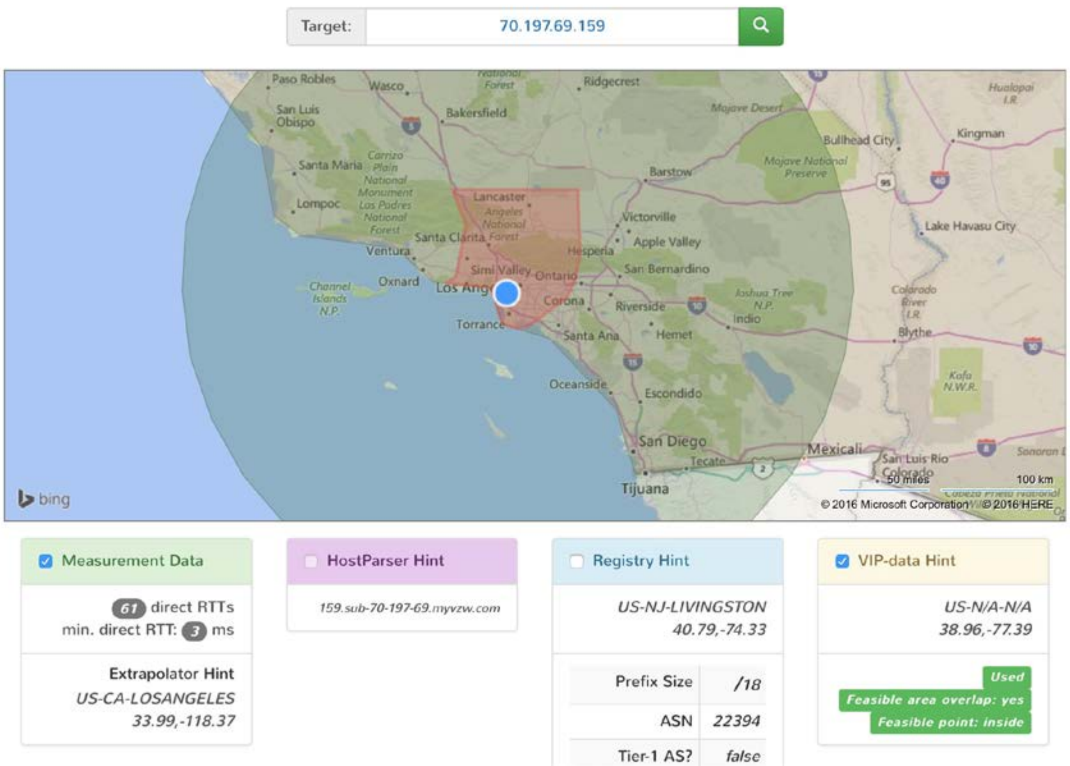


Figure 1: Alidade's Web interface showing the geolocation prediction made for the target IP address 70.197.69.159. The interface also shows the inputs, e.g., VIP-data hint, used for making the prediction. The red polygon in the map represents the feasible region for this target, while the blue marker is the point-based prediction.

2 Methods

Past work on IP geolocation can be loosely categorized into *active* approaches that perform on-demand network measurements to derive constraints on a target's geographic location, and *passive* approaches that rely only on previously collected information to geolocate a target. Both approaches have advantages and disadvantages. Active approaches may be more accurate, but predictions may not be available until new measurements have been taken. Passive approaches can precompute predictions and hence answer queries immediately, without even requiring network access at query time. Importantly, passive approaches are also unobtrusive, and do not risk alerting or annoying the target of a prediction. But passive approaches may not have the target-specific measurement data that would enable better accuracy.

Alidade takes a passive geolocation approach, but Alidade does not rely exclusively on coarse-grained and potentially error-prone data, such as the WHOIS database and hostname-to-location hints. Instead, Alidade filters the hints provided by these data sets by applying constraints derived from large volumes of passively collected network measurements.

In the following sections we examine both active and passive approaches, noting where Alidade borrows techniques.

2.1 Active Approaches

Much of the prior work in geolocating IP addresses relies on on-demand network measurements. *IP2Geo* [24] is an early IP geolocation system that introduces two active IP geolocation techniques. The first technique is *GeoPing*, which requires a deployment of landmarks of known geographic locations that can perform all-pairs latency measurements. To predict the location a target, all landmarks probe the target. *GeoPing* then selects the landmark that has the most similar latency profile (the set of latency measurements from other landmarks) to the user-specified target. It then uses the landmark's location as the prediction for the target. Although this technique is simple and easy to deploy, the location of a target cannot be accurately predicted unless there is a landmark nearby and that landmark has a similar latency profile. At present, Alidade doesn't compile latency profiles or compare the latency profiles of targets and landmarks. The second technique is *GeoTrack*, which performs traceroutes from landmarks to the target to discover routers on the traceroute paths whose DNS names can be interpreted geographically. From this set of routers, GeoTrack locates the target at the closest router's location, where distance is determined in terms of estimated network latency. Alidade's "extrapolator" applies a variation of this technique. By relying only on this relatively incomplete data source, however, GeoTrack's geolocation accuracy is inconsistent.

In contrast to locating the target at the closest landmark or router, Constraint-Based Geoloca- tion (CBG) [14] determines the location of a target by creating circles on the surface of the earth around each landmark, where each circle represents a constraint that bounds the possible location of the target. The size of each circle is a function of the latency between the landmark and target. CBG combines constraints by intersecting the circles, and selects the middle of the intersection as its best estimate of the target's location. One risk in taking this approach is that a single corrupt measurement can lead to an empty intersection. At its core, Alidade is a CBG approach.

Octant [29] builds on CBG by providing a general framework that can combine both positive and negative constraints, that is, information on where the target is likely and unlikely to be, respectively. To handle uncertain or error-prone data sources, Octant combines constraints using a weight-based mechanism that can limit the impact of erroneous measurements. Alidade builds on the Octant framework. In order to process large volumes of measurement data and to geolocate all of the IP address space, Alidade restructures the framework into a parallel Hadoop application so that more memory and compute cycles can be applied.

Topology-Based Geolocation (TBG) [18] uses traceroutes from the landmarks to the target to discover the routers along the network paths and determine inter-router latencies. With this data, TBG performs a global optimization to find a physical placement of the routers and the target that minimizes inconsistencies with the network latencies. By attempting to globally optimize the placement of both the routers and the target, TBG is more sensitive to measurement errors, such as inflated latencies, than constraint-based solutions, where errors tend to be more localized. To some extent Alidade applies this approach too. In particular, Alidade uses all available estimated latencies between pairs of addresses (landmarks, routers, and end hosts) to jointly predict the locations of the routers and end hosts.

Several systems [2, 10, 19, 30] have applied statistical approaches to construct landmark-specific functions that map measured latencies to geographical distances. These systems generally have significant computational requirements, and are currently unable to make use of non-latency-based constraints. *Posit* [9] presents a more recent statistical approach that, while still requiring active measurements, is able to significantly reduce the required number of on-demand probes by precomputing a statistical embedding. At present, Alidade does not construct a sophisticated model of the relationship between latency and distance. Instead, Alidade uniformly assumes that datagrams travel at two-thirds the speed of light, which is very close to the speed of light in optical fiber. Hence, in converting latency to distance, Alidade does not model circuitous fiber paths, nor does it model queuing delays or any other sorts of delays. The resulting constraints tend to be loose, but they are also hard. In particular, provided that no measurements are corrupt and no faster- than-fiber technologies, such as microwave transmission, are employed, the intersection of a set of constraints derived by Alidade from direct latency measurements must contain the actual location of the target. Other work has suggested that if latency is to be converted to distance by a simple multiplicative factor, four-ninths the speed of light might be used. The smaller constant leads to smaller intersection areas, but these areas might be empty or might not contain the target.

Guo et al. [15] propose mining physical addresses displayed on publicly accessible Web sites that are hosted by Web servers with IP addresses in the same prefix as the target address, and using these physical addresses as hints to improve geolocation accuracy and as sources of ground truth to support evaluations. Caruso [6] (as part of the Alidade project) and Wang et al. [28] extend this approach by combining the mined information with latency measurements to offer finer-grained geolocation results. Although these systems produce accurate results in certain experiments, it is difficult to ascertain their actual effectiveness in general. First, it is tricky to determine when an organization is hosting its own Web site. Furthermore, even when an organization does host its own site, for the technique to work the site must list a physical address that is close to that of the hosting location. In previous experiments the best results were obtained when the set of geolocation targets were biased towards belonging to organizations that typically host their own Web servers and publish physical address information on their web pages, e.g., in one experiment reported in [28], university Web servers hosting Web pages listing campus addresses were used as landmarks and PlanetLab nodes were used as targets. Nevertheless, scraped address information from locally- hosted Web sites is a rich source of geographic data, and Alidade includes this information as one of its many data sources.

Gill et al. [13] propose two broad classes of attacks on active measurement-based geolocation approaches. The first misleads geolocation systems by injecting delays to latency probes from specific landmarks at the target, thereby altering the geolocation result by moving the centroid of the constraint intersection in a CBG-based approach. The second targets topology-aware geolocation approaches by altering inter-router latencies in traceroutes, which enables powerful adversaries to place geolocation targets at arbitrary locations. Alidade does not attempt to detect possible adversaries. Unlike active approaches, however, where latency probes can often be easily identified, Alidade also uses a large body of passively collected measurements that piggyback real user TCP connection requests and replies. Adversaries must therefore delay legitimate TCP traffic rather than just latency probes in order to distort much of Alidade's input data.

2.2 Passive Approaches

Although active geolocation approaches can be highly accurate, their dependence on performing on-demand network measurements make them unsuitable for many location-aware applications. Most commercial geolocation systems, such as *MaxMind GeoCity* [21], *EdgeScape* [1], *IPInfoDB* [17], and *HostIP.Info* [22] have instead adopted passive approaches, where they offer their users a pre-computed IP-to-location database that can identify a target's location without additional network access. Unfortunately, the exact methodology for creating these databases are generally proprietary; only the expected accuracy of these databases are typically published. However, the common understanding is that these databases rely on a combination of domain registry information, ISP provided data, host name hints, latency measurements, and other heuristics. Alidade relies on many of the same sources, except that the ISP-supplied ground-truth geolocation data (from one Tier-1 ISP) is used only for evaluation purposes and not as an input to Alidade.

Poese et al. [25] performs an analysis of the accuracy of commericial geolocation databases. They report that while geolocation databases are extremely accurate at the country level, they perform poorly at the city level. Note that Poese et al. did not analyze EdgeScape (or Alidade).

In addition to GeoPing and GeoTrack, IP2Geo [24] also introduces *GeoCluster*, a passive ap- proach that partitions the IP address space into geographically co-located clusters. GeoCluster then assigns each cluster to a geographic location based on the geographic information extracted from user registration and usage databases. The effectiveness of this approach is largely limited by the availability of such databases, the geographic coverage of the users in the databases, and the accuracy and freshness of the self-reported user location information. At present, no such data is available to us, but if it were, it could be used as an input to Alidade.

3 Assumptions and Procedure

3.1 IP geolocation should not remain a black-box system.

Some of our recent efforts have been towards exposing the internals of the IP geolocation techniques employed by Alidade to the users of the system. When a geolocation system makes a wrong prediction for a target IP address, there is, often, little or no explanation for the incorrect prediction. By exposing the internals—details on inputs that were available for making the prediction—a geolocation system can provide valuable hints on why the prediction failed and what alternatives are available. The use cases of applications that require geolocation predictions vary across a huge spectrum: some applications need a system that provides without fail a latitude-longitude coordinate for every IP address, regardless of what data sources were used in generating that prediction; on the end of the spectrum are applications that require predictions in the form of a polygonal region with the guarantee that the region includes all possible locations of the IP address. Alidade offers extensive details, e.g., what inputs were available for making a prediction, what set of inputs were ignored and why, and allows the users to make an informed choice on how to make use of its predictions. The following are a few examples of predictions made by Alidade with details on what inputs were used or ignored.

Conflicting Hints. Figure 2 shows a geolocation prediction made by Alidade along with details on the inputs used for making the prediction. Observe that while a registry hint is available for the target IP address, it was not used in making the prediction. The registry hint, which points at Livingston, NJ conflicts with both the extrapolator hint (pointing at Los Angeles, CA) and the region formed by intersecting delay-based measurements to the target.

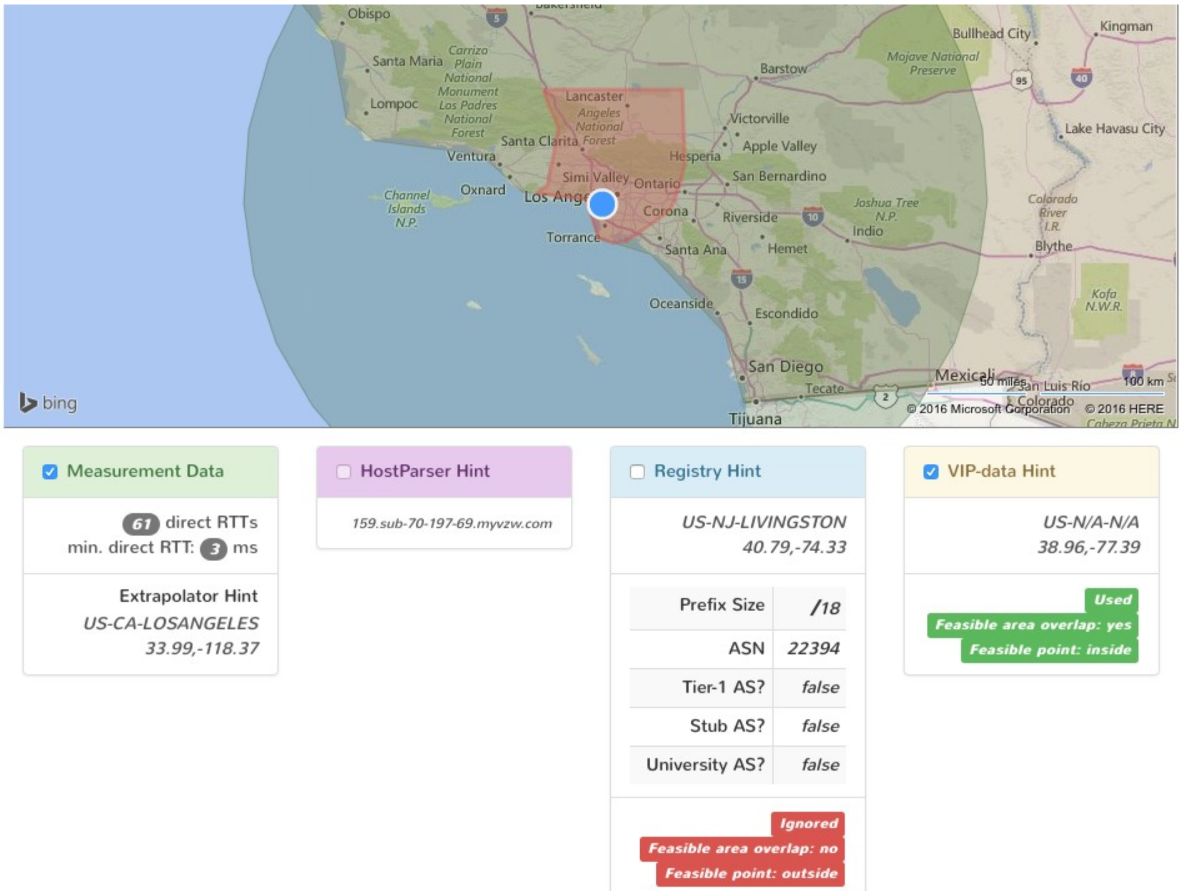


Figure 2: Prediction made from delay-based measurements and VIP-data. Registry hint, pointing at Livingston, NJ was ignored since it conflicts with the region computed intersecting direct measurements.

Incorrect Ground Truth. It is hard to evaluate geolocation systems since there are not many publicly-available ground truth location data for IP addresses. Our preliminary evaluations, more- over, indicate that such data sets may already have been incorporated into commercial geolocation databases (for instance, in form of "hard-coded" IP-to-location mappings). Errors in the ground truth location data further make comparative evaluations harder to perform.

Figure 3, for instance, shows a prediction where Alidade alerts the user (in the last step of the gist shown in the figure) about the incorrect ground truth data associated with the concerned IP address. The distance between the point-based estimate generated by Alidade and the ground truth location is referred to as *error distance* and provides an estimate of the inaccuracy of the geolocation prediction. Trusting the ground truth data blindly will have resulted in the user concluding that Alidade's estimate is incorrect by over 2500 km whereas the true location is most likely to be within 20 km distance of Alidade’s point-based estimate; the difference of over two order of magnitudes between the error distance values could spell the difference between the best and worst geolocation predictions.

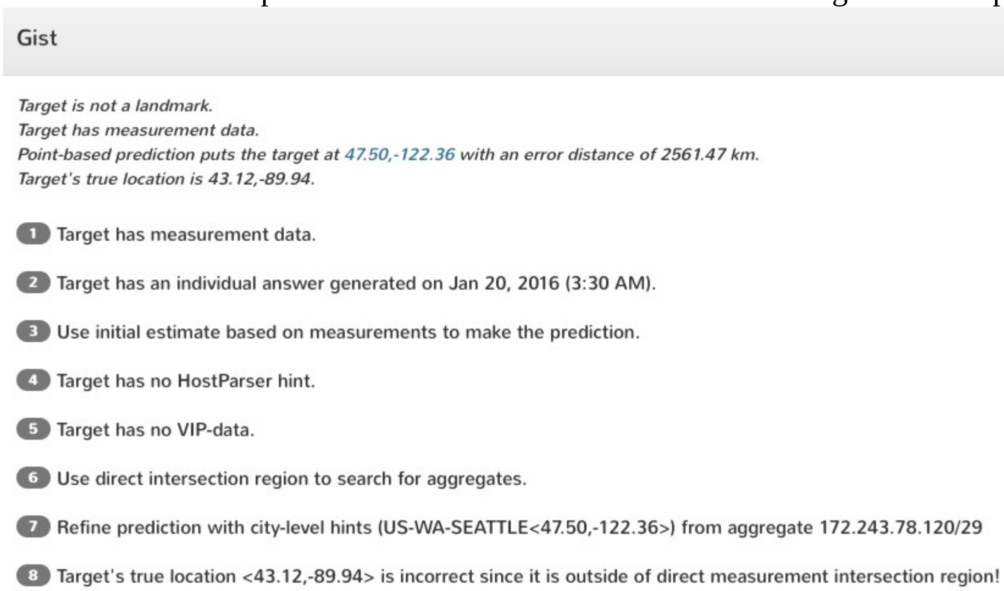


Figure 3: Alidade’s prediction for a target highlighting (in the last step of the gist) that the ground truth location of the target is incorrect.

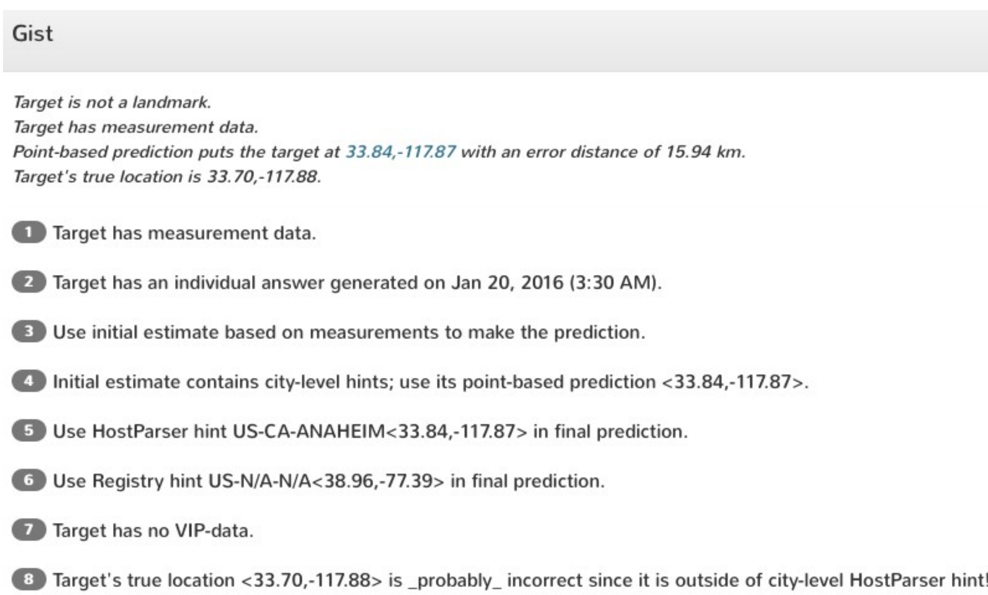


Figure 4: Alidade’s prediction for a target highlighting (in the last step of the gist) that the ground truth location of the target may be incorrect. The ground truth location is contained in the area formed by intersecting direct measurements, but not contained within the feasible region of the prediction.

Figure 4 shows a prediction where Alidade indicates that the ground truth location data could potentially be incorrect. Alidade treats city-level HostParser hints as a highly reliable data source and the ground truth location, in this example, conflicts with the city-level HostParser hint available for the target. Since the HostParser hint is still a non-measurement-based data source, Alidade marks the ground truth location data as *probably* incorrect.

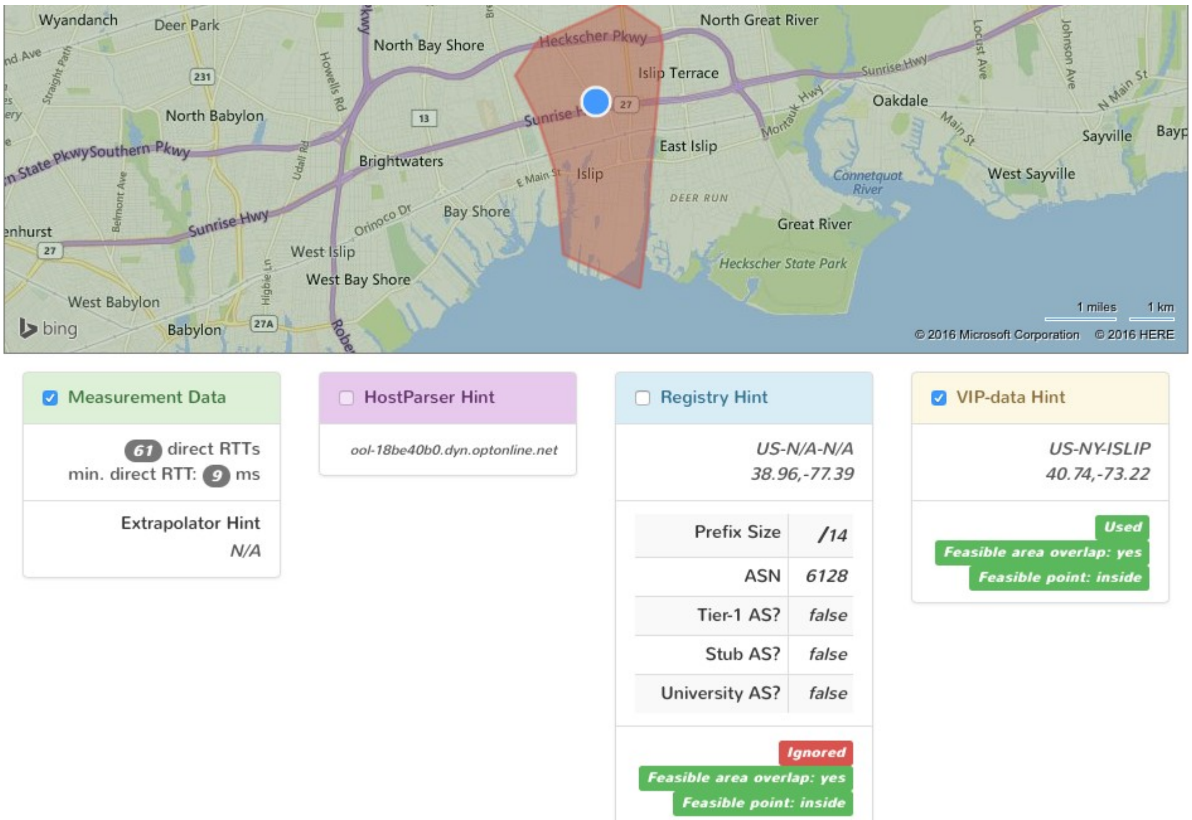


Figure 5: Alidade's prediction for a target highlighting an unused Registry hint. The interface indicates that even though the hint was not used in making the prediction, it can be included to further increase the user's confidence in the prediction.

Unused Hints. Regardless of what hints are available for a target IP address, Alidade may choose simply a subset of the hints to make a geolocation prediction. The system, for instance, can ignore a Registry hint if a VIP-data hint is available at city-level (Figure 5); even if the registry hint agrees with the VIP-data hint, it is marked as redundant. Alidade, however, passes this information to the user who may consider adding the Registry hint as well to the prediction, perhaps to increase the credibility of the prediction.

3.2 Use of city-level and state-level shape files.

Alidade uses simplified versions of high-resolution shapes of cities when converting location hints into polygonal constraints. The simplification process involves running the α-shapes algorithm to remove vertices such that area of the simplified polygon only adds to but does not take any away from the area of the original (high-resolution) polygon. Figure 6, for instance, shows Alidade making use of the simplified shape available for Chicago, IL. Alidade, currently, has support for using shape files for city-level hints from any data source. We are also searching for good sources of shape files for major cities in countries other than the United States.

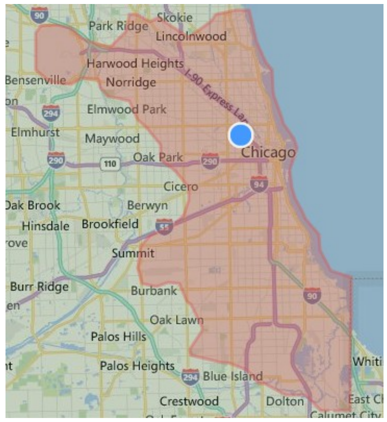


Figure 6: A geolocation prediction making use of a simplified shape file for Chicago, IL.

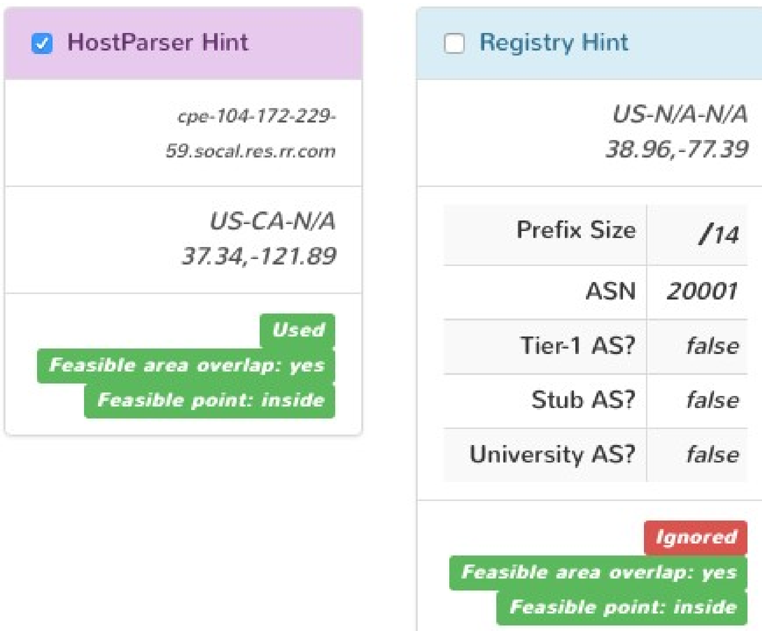


Figure 7: Use of state-level hint in making a geolocation prediction. The system makes use of a state-level HostParser hint and ignores a country-level in making the prediction.

We also have a significant number of region-level or state-level hints from HostParser and Registry, and the system has support for leveraging these hints in making better predictions. Figure 7, for instance, shows the system using a state-level HostParser hint, US-CA-N/A and ignoring, consequently, a redundant country-level hint from Registry. We recently added support for using shape files for state-level hints and will soon measure the impact of this change on geolocation accuracy.

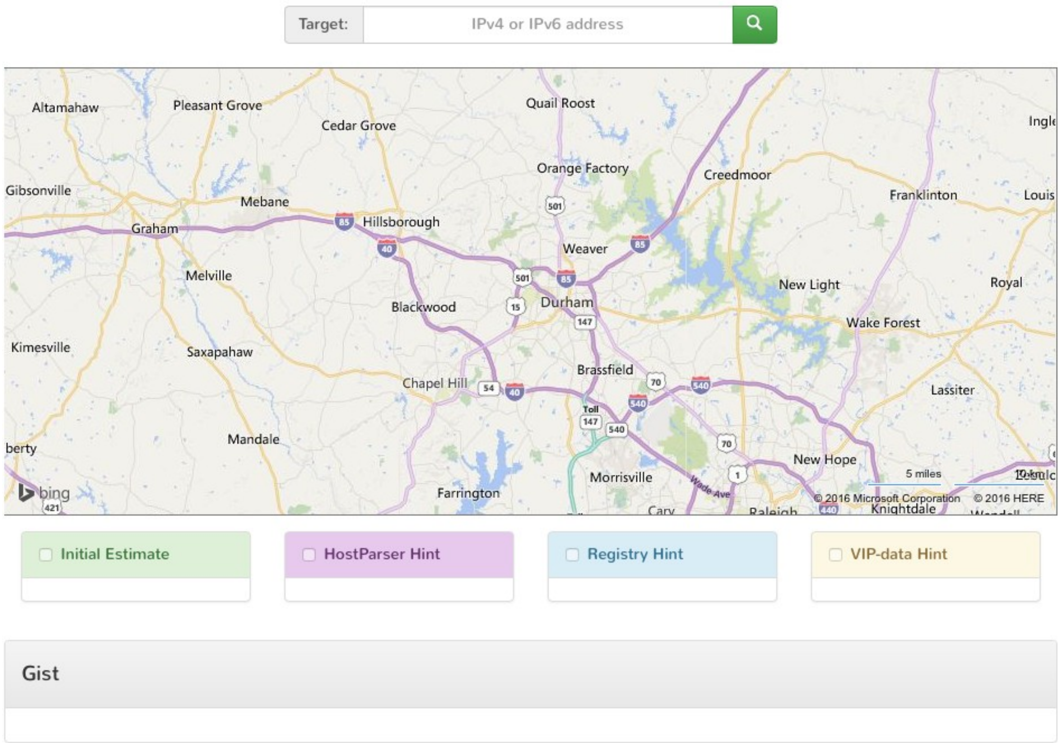


Figure 8: Alidade's new Web interface for looking up geolocation predictions. Once the details of the predictions for the IP address specified by the user is retrieved, the map is updated with the feasible region and point-based predictions. The panels below the map are also populated with the relevant hint details, if available. The gist section is updated with the sequence of steps that Alidade followed to make the prediction.

3.3 Improved User Interface.

Alidade's geolocation predictions are often accompanied with extensive information on the data sources used and a sequence of steps followed to generate the prediction. A clean interface, hence, is key to visualize the different pieces of information and help the user understand how Alidade made the prediction and highlight the most important data used in making the prediction. To this end, we made significant improvements to the Web interface and the screenshots of this interface highlighting various features are shown in Figures [8, 9, 10, 11, 12].

3.4 Evaluation.

We evaluated Alidade using a set of 27493 IP addresses for which we have ground truth location information. The data was obtained from a set of *SpeedTest* servers that are used by end users to measure the speed of their Internet connection. The servers log the IP address and the location data (typically, in the form of a latitude-longitude) reported by the browser or application used by the end users to run the speed test. We carefully selected only those IP addresses where the test was performed from an iPhone and over a WiFi Internet connection. We dropped IP addresses that had repeat measurements but where the measurements were still associated with the same location. The rationale behind the heuristic is that if the location data reported by the phone was indeed based on GPS signal, it is unlikely to the same over a set of repeated measurements. It is, nevertheless, still possible that the data set contains IP-address-to-location mappings where the location data was not reported by the phone but was retrieved by the SpeedTest server using a commercial geolocation service. The data set, unfortunately, does not indicate the exact source of location data.

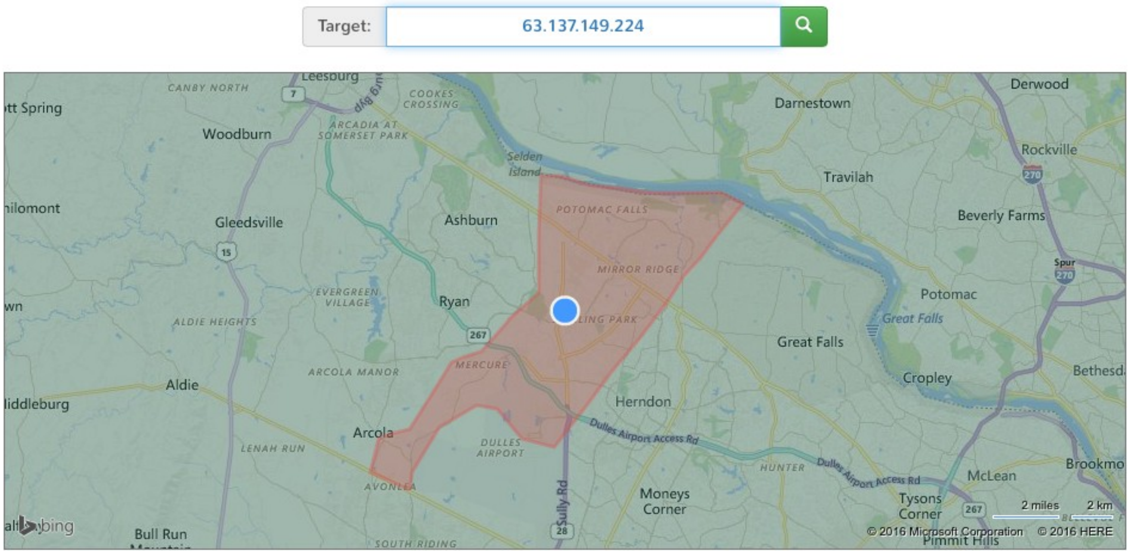
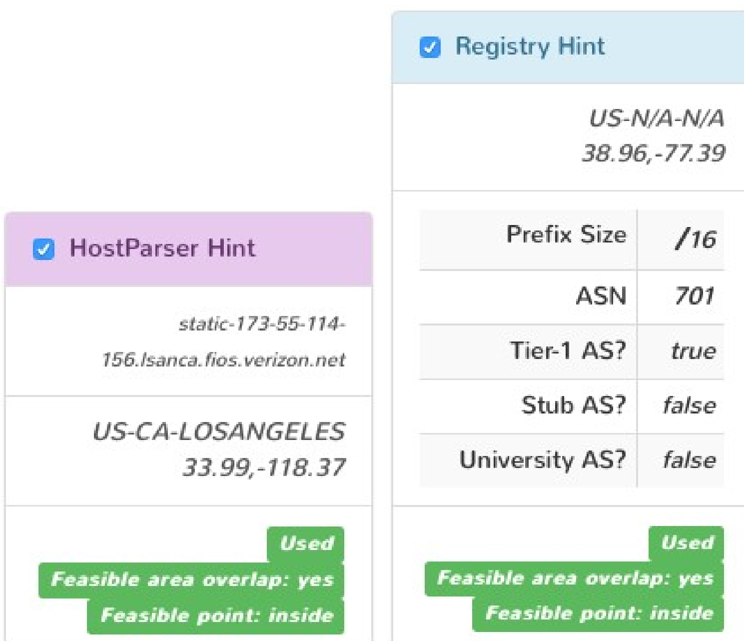
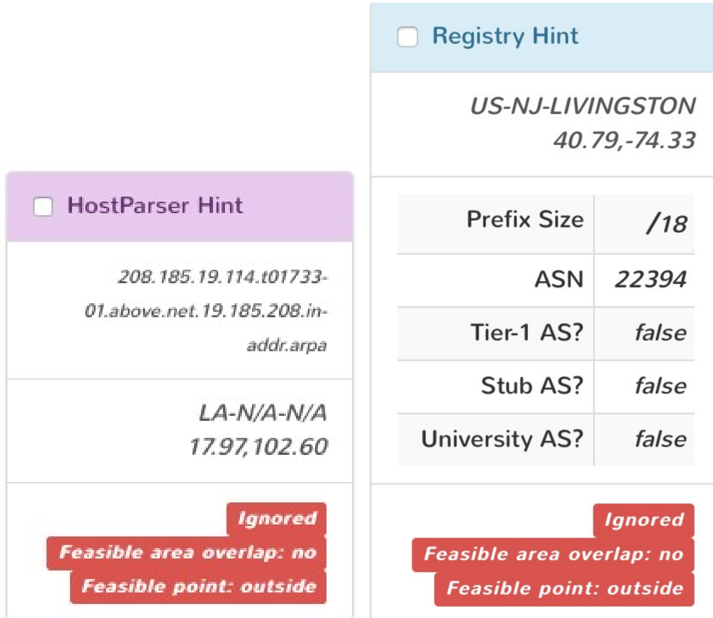


Figure 9: The Web interface showing the map updated with the feasible region (in red) and centered around the point-based estimate (in blue) after the user queried the system for geolocating the IP address 63.137.149.224.



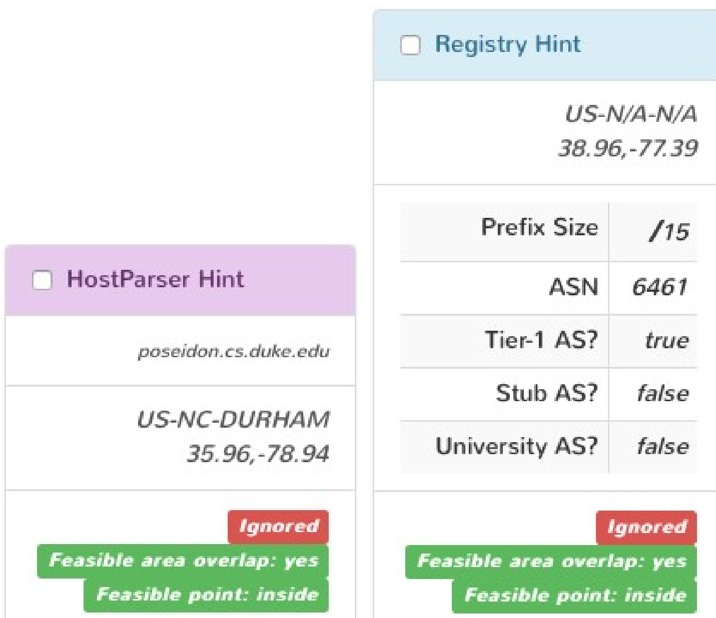
(a) *HostParser Hint.* (b) *Registry hint.*

Figure 10: Details of HostParser and Registry hint used in making a geolocation prediction.



(a) *Conflicting HostParser Hint.* (b) *Conflicting Registry hint.*

Figure 11: Details of conflicting HostParser and Registry hints. Alidade indicates the source of the conflict: either the hint area did not overlap with the feasible region, or the point-based prediction was outside the hint area, or perhaps, both.



(a) *Ignored HostParser Hint.* (b) *Ignored Registry hint.*

Figure 12: Details of unused HostParser and Registry hints. Sometimes a hint is ignored because a better input for making the prediction is available. In such cases, Alidade marks the hint as ignored and indicates that the hint could have been used in making the prediction, but was unnecessary.

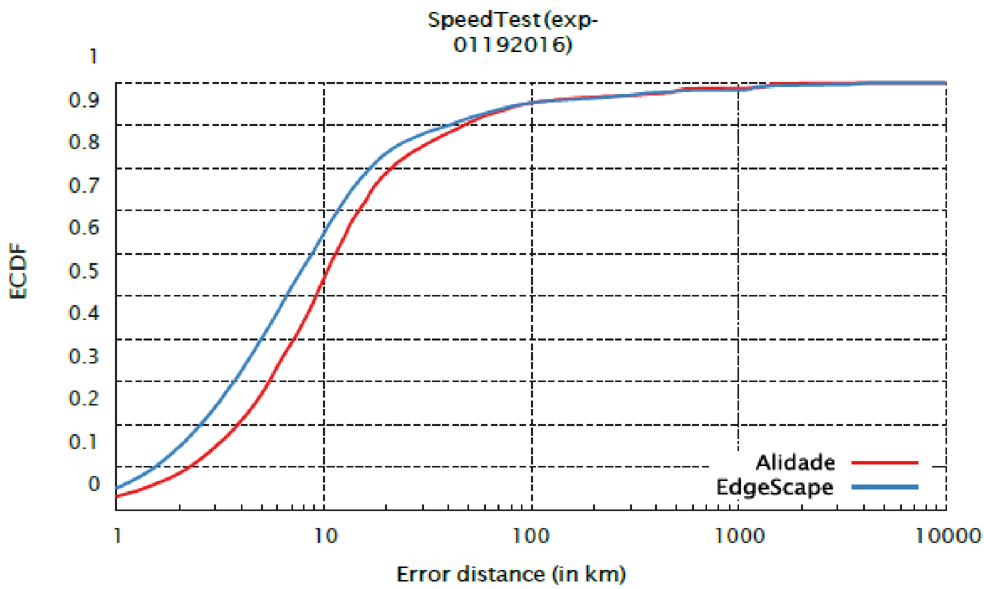


Figure 13: Alidade vs EdgeScape: comparison of geolocation accuracy using the SpeedTest data set. Although the X-axis (error distance, in km) starts at 1 km, given the accuracy of ground truth data and rounding errors associated with latitude-longitude coordinates, we ignore error distances below 10 km.

Figure 13 compares the accuracy of Alidade and EdgeScape [1] in geolocating the targets from the SpeedTest data set, described above. The X-axis represents the error distance, in km, which is defined as the distance between the true location of a target and the point-based prediction made by a geolocation database. Although, the X-axis starts at 1 km, given the accuracy of ground trust data and rounding errors associated with latitude-longitude coordinates, we ignore error distances below 10 km. In other words, if the error distance associated with the prediction for a target is below 10 km, we treat is as having no error in geolocation. Overall, EdgeScape performs slightly better than Alidade: nearly 65% of the predictions made by EdgeScape have an error distance of 10 km or less compared to 55% of that made by Alidade. The tail portion of the ECDF show a slightly different behavior, with the maximum error distance associated with Alidade's geolocation predictions being approximately 4000 km while that of EdgeScape's being over 12000 km. We have identified a few new heuristics and optimization to improve Alidade's predictions further and perhaps, these suffice to help Alidade provide better geolocation accuracy compared to EdgeScape.

3.5 Public Access.

The following people have made use of the geolocation predictions from Alidade in some form in their work: Prof. Philippa Gill, Stony Brook University; Prof. Reza Rejaie, University of Oregon.

4 Results of relating latency to distance

One of the discouraging findings of the Alidade project is that the measured network latencies that are available to us are too large to provide small geometric constraints. Furthermore, the relationship between measured latency and physical distance is not as clearcut as might be imagined. Over medium and long distances, almost all traffic on the Internet is carried over optical fiber buried in fiber conduits. The conduits often do not provide line-of-sight paths between pairs of network hosts, so distance estimates based on latency may be overestimates. More subtly, while it would be convenient if latency was simply the length of the fiber conduit divided by the speed of light in fiber, our experiences led us to believe that in practice latencies are typically larger. So in the final year of the project, we devoted a great deal of effort to trying to understand this relationship.

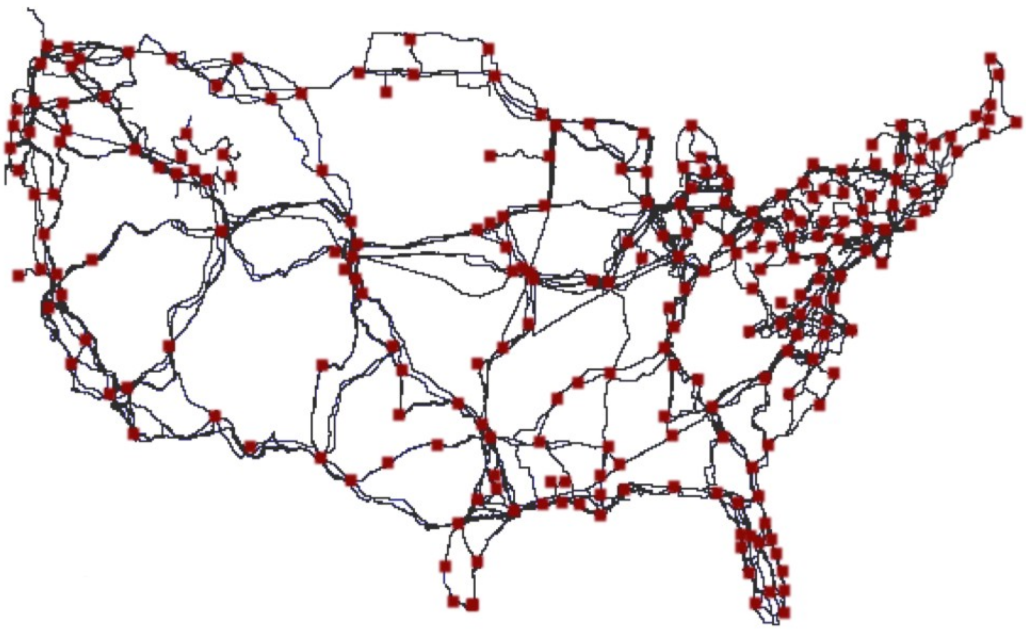


Figure 14: Fiber Conduits in USA

4.1 The "InterTubes" map

Recently Durairajan et al. analyzed the fiber-optics infrastructure of the contingent USA [8]. The authors compiled and published a fiber map (the "InterTubes" map) using publicly available data of ISPs and other public records. Published fiber map is larger than the networks of largest ISPs, as it is a compilation of fiber links belonging to 20 different ISPs. One finding of the paper is that conduit sharing among ISPs is common, as laying out new fiber is expensive. The authors also compared the conduit lengths to road distances. The fiber map, which is shown in figure 14 has 273 nodes and 540 links [1].

[1]Copyright of the image belongs to Durairajan et al.

We obtained conduit lengths from the authors. For 31 links, the authors were not able to obtain the actual length of the conduit, so the line of sight distance is provided instead. Figure 15 shows the comparison of conduit lengths to line-of-sight distances between the endpoints of each conduit. The median conduit length is 167 km and about 2/3 of the conduits are longer 100 km. The CDFs shown in figure 15a show that the distributions of conduit lengths and line-of-sight distances are very close to each other. Figure 15b shows the CDF of the ratio of conduit length to line-of-sight distance between its endpoints, and we see that the median and 95th percentile of this ratio are 1.2 and 1.28 respectively.

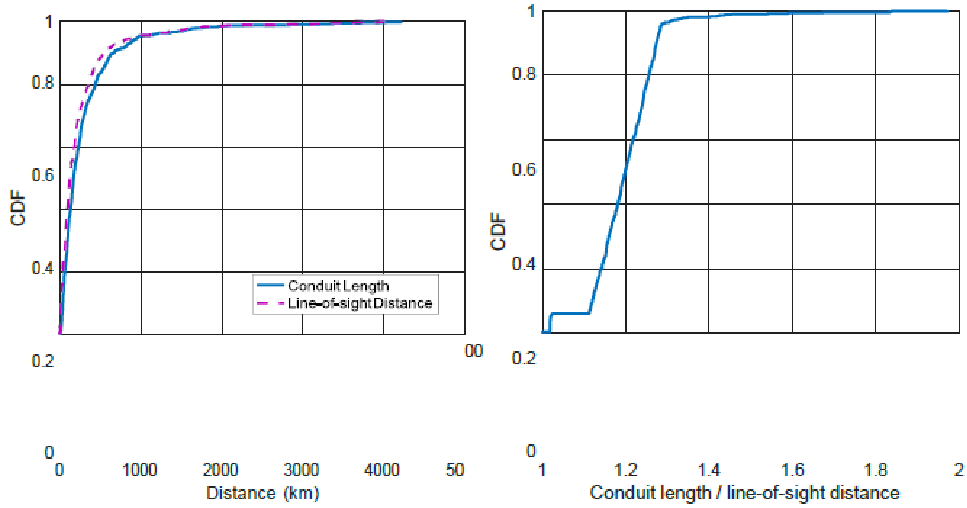
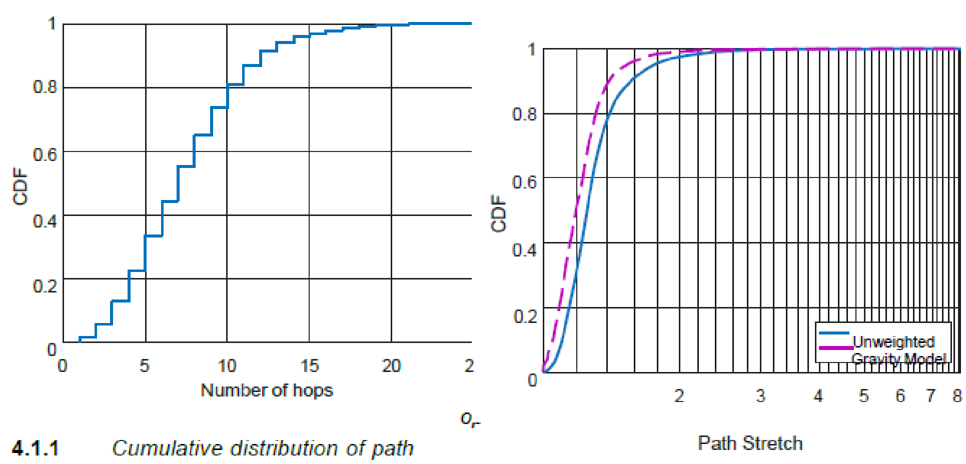


Figure 15: Comparison of conduit lengths to line-of-sight distances

Since individual link lengths are close to line-of-sight distances, we wanted to examine how the shortest path lengths between all pairs of cities would compare to line-of-sight distances.



4.1.1 *Cumulative distribution of path lengths (h count)*

4.1.2 *Cumulative distribution of path stretch*

Figure 16: Results of all-pairs-shortest-path computation

For this purpose, we computed all-pair shortest paths between 248 cities which have population at least 20.000.[2] Then we computed the *stretch* for all pairs of cities, which is the ratio of the length of the shortest path in the fiber map between two cities to the shortest distance between the

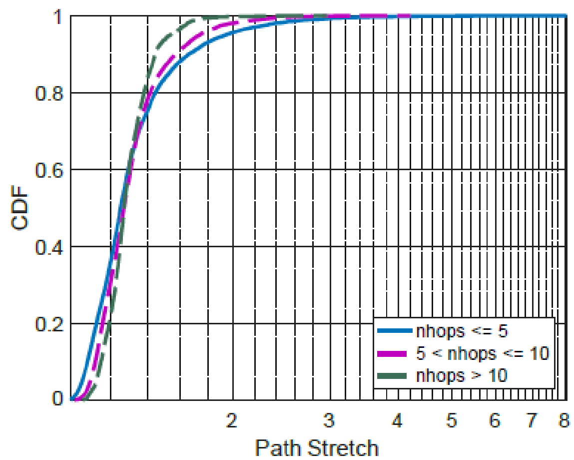


Figure 17: Stretch distributions for different path lengths(number of hops)

same cities on the surface of the Earth. Figure 16 shows the distribution of the path lengths (in terms of hop counts) and the distribution of all-pairs stretch under two different scenarios. We see that the median and the 95th percentile path lengths are 7 and 14 hops respectively. The stretch results are more encouraging. The green line, marked as unweighted stretch, shows the cumulative distribution of this value. Median and 95th percentile stretch values are found as 1.27 and 1.75 respectively.[3]

The red line shows the distribution of the stretch when we used the so called gravity model which estimates the traffic between two cities as proportional to the product of their populations. With gravity model, the results get even better, and median stretch is found as 1.2, and on the 95th percentile stretch is found as 1.53.[4]

[2]The populations were obtained from 2013 Census data.

[3]The stretch values reported in this section do not take the speed-of-light in fiber into account, which is roughly 2/3 of speed-of-light in vacuum.

[4]When we don't ignore cities with population less than 20, 000 and examine stretch for all pairs of cities regardless of population, the computed values are slightly higher. In the unweighted case, median and 95TH stretch values are 1.32 and 1.86 respectively. When we use the gravity model, median and 95th stretch values are found as 1.26 and 1.56 respectively.

Even though the median and even the 95th percentile stretch values are not very high, it is interesting to examine the relationship between stretch and the number of hops for each city pair. One might conjecture that higher stretch values in the tail are caused by long circuitous paths. Figure 17 shows the stretch distribution for three different intervals for the number of hops: less than 5, between 5 and 10, and, more than 10. The result is surprising. For the small values of stretch, the distributions support this claim, as larger number of hops correlate with higher stretch. However, the curves meet around 60th percentile, and for the higher stretch values in the tail, we see the opposite. This is probably due to some paths having a large number of intermediate cities along a relatively straight path, with little impact on stretch.

4.2 Examining fiber conduits of ISPs individually

After examining the entire dataset, we next examine the networks of each ISP in isolation. The InterTubes dataset lists all the ISPs using a conduit, and our analysis here is based on this information. Note that in reality at least some ISPs probably have more links in their backbone networks. This might be due to either some long-haul links not being included in the InterTubes dataset or the list of ISPs using a particular link in the dataset being incomplete. Link sharing is very common and some links are very heavily shared as reported in [8]. This is also very clear from the sizes of networks as we report in Table 1. The median and 95th percentile values of conduit length to line of sight distance ratio for each ISP is very close to the corresponding values for the entire dataset. This holds true even for ISPs which have a small number of long-haul links.

EarthLink and Level3 have the largest number of fiber conduits in their backbones, each being listed on around 85% of all the conduits provided in the dataset. Thus, it is not surprising that the median and 95th percentile stretch values for all pairs shortest paths for these two networks are very close to the corresponding values for the entire dataset. For networks with smaller number of fiber conduits, both median and 95th percentile stretch values are larger[5]. Moreover, almost for all ISPs using the gravity model (i.e. traffic between two cities is assumed proportional to the product of their populations) results in a smaller stretch compared to the unweighted case.

Table 2 also shows the median and 95-th percentile stretch values for the 9 largest ISPs, each of which has long-haul links connecting over 100 locations. The median stretch across these networks is smaller than that reported in earlier work [5] analyzing research networks such as Internet2. Note that the values in Table 2 are not directly comparable to the stretch values we compute above for the entire set of 273 locations – no single ISP covers all locations, with the largest covering 248 – but even when covering fewer locations, individual ISP networks show higher stretch. Further, a recent measurement study [5] has reported end-to-end latencies over default routes between well-connected locations as 3-4$\times$ inflated over c-latency. Thus, being able to aggregate and utilize the entirety of this fiber infrastructure could reduce latencies by as much as 50%. In the following, we estimate the performance and cost of such a network connecting large population centers.

[5]In Table 1 we appended an $*$ to some ISP names. For these ISPs, the network consisting of all the links that they are listed as using is not connected. The stretch computations were done over pairs of connected endpoints for these ISPs. This might be one reason for higher stretch values for some of these ISPs.

5 Using a CDN to measure conduit latencies

We located the servers of a major CDN within 25 kilometers radius of each conduit endpoint. We were able to find servers near endpoints of 51% of the conduits. When considered as a directed network, the fiber map has 1080 links, and we found servers near conduit endpoints for 552 of these links. At some locations, the CDN has many clusters, utilizing multiple ISPs for connectivity. For each conduit with CDN clusters on both of its endpoints, we ran traceroutes between all pairs of clusters near two endpoints of the conduit. Figure 18 shows a schematic description, where we have 2 clusters on one end and one in the other, resulting in 4 different traceroutes along this fiber conduit in two directions. Basically, we wanted to find out for how many fiber conduits we would find measurements with RTT close to c-latency, i.e. the time it would take light to travel round-trip in the fiber conduit. Figure 19 shows the measurements that we ran on the USA map. Red dots show the conduit endpoints and the green dots show the nearby CDN clusters. Any measurement between a pair of CDN clusters is marked with a line between them. Some lines are thicker, because there is a large number of clusters near some conduit endpoints, resulting in a large number of pairwise measurements, with one line for each drawn, sometimes between same physical locations.

Table 1: Network size, link length and stretch analysis of individual ISPs

					All Pairs Stretch			
	Network Size		Conduit Length		Unweighted		Gravity Model	
ISP	nodes	links	Median	95^{th}p.	Median	95^{th}p.	Median	95^{th}p.
EarthLink	248	468	1.20	1.31	1.35	1.83	1.32	1.66
Level3	245	456	1.20	1.29	1.36	1.83	1.33	1.64
Comcast	149	195	1.20	1.28	1.48	2.11	1.45	1.85
CenturyLink	145	181	1.20	1.34	1.55	3.37	1.55	3.23
TWC*	129	183	1.21	1.42	1.48	2.03	1.44	1.90
Verizon	116	151	1.19	1.29	1.54	2.16	1.53	1.93
AT&T	107	143	1.20	1.29	1.43	2.66	1.40	2.19
Sprint*	106	118	1.21	1.30	1.75	5.47	1.59	5.12
HE*	105	131	1.21	1.47	1.45	2.67	1.42	2.45
Tinet	98	122	1.22	1.47	1.47	2.14	1.43	1.97
NTT	98	116	1.20	1.40	1.58	3.76	1.64	3.99
Zayo*	98	111	1.21	1.45	1.47	2.28	1.43	1.94
XO*	96	111	1.19	1.28	1.68	4.35	1.65	3.87
Cogent*	93	98	1.20	1.34	1.59	2.90	1.53	2.85
TeliaSonera	87	100	1.20	1.35	1.56	2.93	1.48	2.82
Cox	79	97	1.20	1.47	1.52	6.79	1.46	5.03
Tata *	71	81	1.20	1.47	1.49	3.34	1.44	3.03
DeutscheTelekom	56	62	1.19	1.33	1.62	5.85	1.51	5.44
Integra	53	66	1.22	1.47	1.41	3.01	1.41	2.36
SuddenLink*	39	42	1.19	1.31	1.49	3.14	1.35	3.88

Note that the clusters are chosen such that they are within 25 km of conduit endpoints. So, there will be an additional delay for covering this distance and extra switching delay per hop. Moreover, in some areas the provided conduit endpoint location is a central point between multiple conduit endpoints inside the same city, adding further uncertainty to the additional latency overhead. For these reasons, we allowed some deviation from the target round-trip-time along the fiber links.

When two directions are considered separately, we identified 111, 000 server pairs for running traceroutes. 5022 of these measurements are between servers in the same network. Discarding pairs which didn't reach the target or with no measurements performed due to other problems, we have 3, 800 measurements with both servers in the same network, and over 90, 000 measurements with servers in different networks. Some of these measurements are for maintaining connectivity within a city i.e. not along any fiber conduit.

Considering only the measurements along the fiber conduits, the numbers fall to 2, 960 measurements with both servers in the same network (along 125 unique directed fiber links), and, over 80, 000 measurements between servers in different networks (along 543 unique directed fiber links). When two clusters are getting connectivity from the same provider, the median minimum RTT is 54% higher than *c-latency*. For other pairs of clusters, median minimum RTT is 93% higher than *c-latency*. However, since the number of available measurements between clusters getting network connectivity from different providers is much larger, most of the conduits with measured RTTs close to *c-latency* come from this set of measurements.

Table 2: Stretch values for all pairs shortest paths

ISP	Median	95th. prc.
EarthLink	1.98	2.68
Level3	1.99	2.68
Comcast	2.17	3.09
CenturyLink	2.27	4.94
TWC	2.17	2.98
Verizon	2.26	3.17
AT&T	2.10	3.90
Sprint	2.56	8.02
HE	2.13	3.91

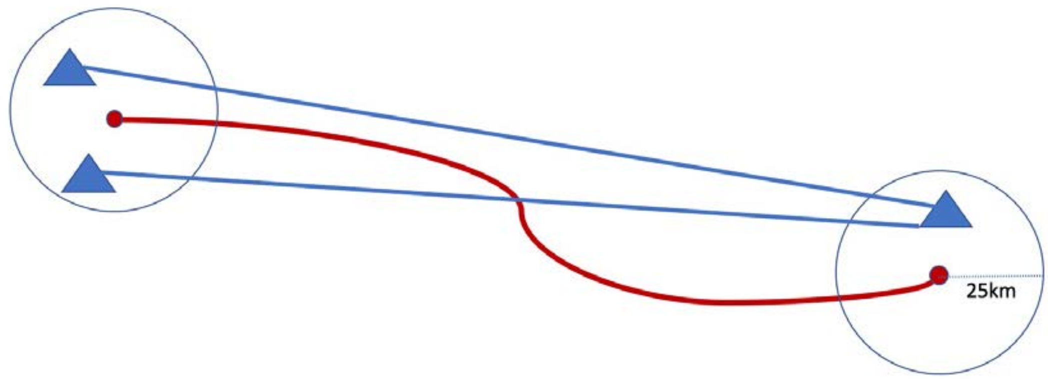


Figure 18: A schematic description of measurements performed between servers of the CDN along a fiber conduit. Blue triangles denote CDN clusters(servers) within 25 km of a conduit endpoint, and the lines between CDN clusters denote a traceroute run is between the two.

As we mentioned above, we allowed some deviation in the actual RTTs from *c-latency* to count one measurement as going over the fiber link. 124 fiber links (out of 1,080) have RTTs within %25 or 0.5 ms of *c-latency*, corresponding to 11% of all links[6]. When we allow 40% or 0.5 ms deviation,we have 182 fiber links covered. This coverage is lower than we expected.

We tried to verify whether the traceroutes followed the fiber conduits by geolocating the routers. Even though examining router locations cannot prove that any traceroute went over the intended fiber conduit, it can prove that it did not. With this process we were able to identify traceroutes which did not follow the target fiber conduit, and hence resulted in high RTTs due to the longer BGP paths. We used a heuristic to potentially identify traceroutes that followed the intended fiber conduit. After geolocating the routers, we identified the pair of routers ($r1$, $r2$) with the maximum line-of-sight distance among all consecutive routers in the traceroute. Traceroutes in which $r1$ and $r2$ are within a small distance of the conduit endpoints are possible candidates for measurements which followed the fiber conduits. When we limit ourselves to a single traceroute with the minimum RTT among all server pairs for each conduit and use the described heuristic, we end up with 84 traceroutes along different conduits. Even for these traceroutes the median inflation of RTT over $c-latency$ with respect to length of the fiber conduit is 1.59. One of many similar examples with a high RTT the reason of which we cannot pinpoint is for the fiber link between Las Vegas and Dallas. This link is 2105.95 km long, resulting in a 20.65 ms RTT inside the fiber. Minimum measured RTT is 30.58 ms which is 48% higher than the lower bound. The traceroutes are 4 hops long, the first hop is in Las Vegas within 1 km of the conduit endpoint and the second hop is in Dallas, also within 1 km of the conduit endpoint. We have 30 traceroutes between the server pair with this minimum RTT on two separate days with 3 hours between the measurements, so congestion is probably not the problem. In the remainder of this document we describe our further efforts to verify these results through comparison to other sources of latency data and fiber conduit lengths from other networks.

[6] 0.5 ms is the time for light to travel round-trip in 50 km in fiber.

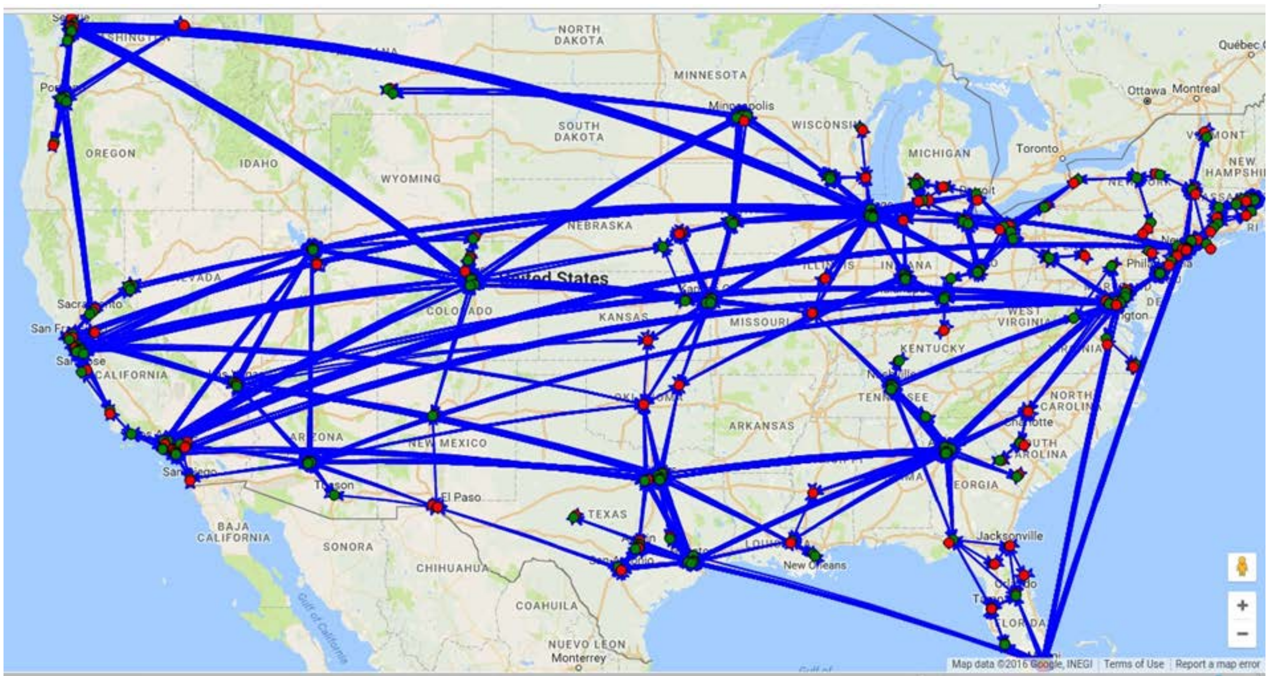


Figure 19: The measurements we performed between the clusters of the CDN.

6 Latency Analysis for AT&T

AT&T publishes latencies between major cities it connects in USA on its Web site [3]. Every 15 minutes the data is updated. We have collected this data between February 27, 2017 and April 7, 2017 by recording it every 30 minutes. Our goal in collecting this data is to compare the latencies published by AT&T to our measurements between the same city pairs, and also to the physical limit imposed by speed-of-light in fiber.

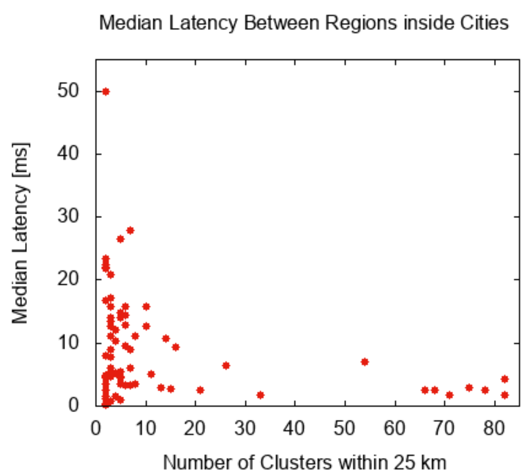


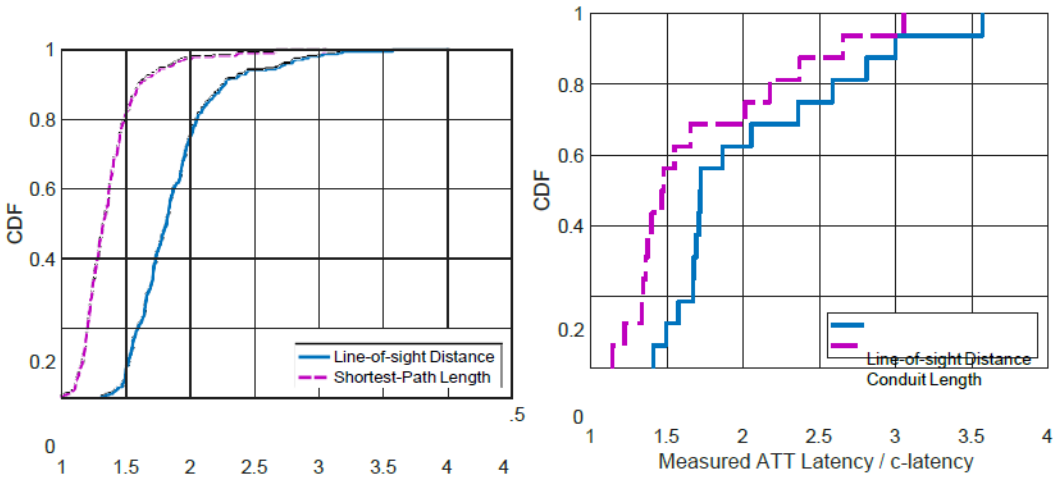
Figure 20: Median latency between clusters in each city with respect to the number of clusters in each city.

6.1 Latency Inflation in AT&T's Network

For each city pair in the published data, we computed the ratio of the AT&T latency to the round- trip time for light to travel the line-of-sight distance between the city pair inside fiber. The blue line in figure 21a shows the CDF of this ratio. We see that this ratio varies between 1.31 and 4.0 and the median value is 1.82. The dotted purple line shows the same ratio when we use the shortest path length between the two cities (computed from the InterTubes dataset utilizing all the links marked as used/owned by AT&T) as the baseline instead of the line-of-sight distance. Since we take the length of the fiber conduits into account, the purple line lies strictly to the left of the blue line, with minimum, median and maximum values being 1.00,1.33 and 3.05.

The computation of the shortest paths revealed the existence of some links in AT&T's backbone not included in the InterTubes dataset. This is because when we did this computation for the 259 city pairs utilizing just AT&T links according to the InterTubes dataset, we identified city pairs for which the published latency violated the speed-of-light constraint. This can only happen if there are links in AT&T's backbone which do not exist in the InterTubes dataset, or even if the link is in the dataset, it is not marked as used/owned by AT&T. For example, in the dataset there is a fiber conduit between Chicago and Indianapolis, with length 322.8 km, but this conduit is not marked as used by AT&T. Over this fiber conduit, the round-trip travel time of light is 3.16 ms. However, the shortest path with AT&T's links is found as the 927.86 km long *Chicago-Springfield- St.Louis-Indianapolis* path, resulting in a lower bound of 9.1 ms round-trip travel time. However, the minimum latency published by AT&T is 5 ms between Chicago and Indianapolis, illuminating the problem with the InterTubes dataset. Note that even if we assume the direct link between Chicago and Indianapolis is in AT&T's backbone, resulting latency inflation would be 58% which is more than what we would expect. We have identified 7 city pairs (out of 259 for which we have latency data) for which the shortest paths found using AT&T's links in the InterTubes dataset result in speed-of-light violations based on the published RTTs.

The fact that some of AT&T's links are missing is probably causing some inflation values reported above and plotted in figure 21a to be lower than they really are. Missing links increase the shortest path length for some city pairs, lowering the computed inflation and masking the real one. That's why we probably see a minimum inflation of 1.00 (since we also removed 7 city pairs with inflation values <1 implying speed-of-light violations). This explanation is also supported by the CDFs plotted in figure 21b for the city pairs with a direct fiber conduit between them. For the corresponding purple line, the minimum, median and maximum of the inflation over *c*-latency is found as 1.14,1.47 and 3.05 respectively. Figure 22 provides further proof of this, since some of the largest inflations are observed for city pairs with a direct link between them, and some very low inflations are between cities 5 and 7 hops away.



6.1.1 *All City Pairs*

6.1.2 *City Pairs connected with a direct fiber con- duit*

Figure 21: Inflation in AT&T's network

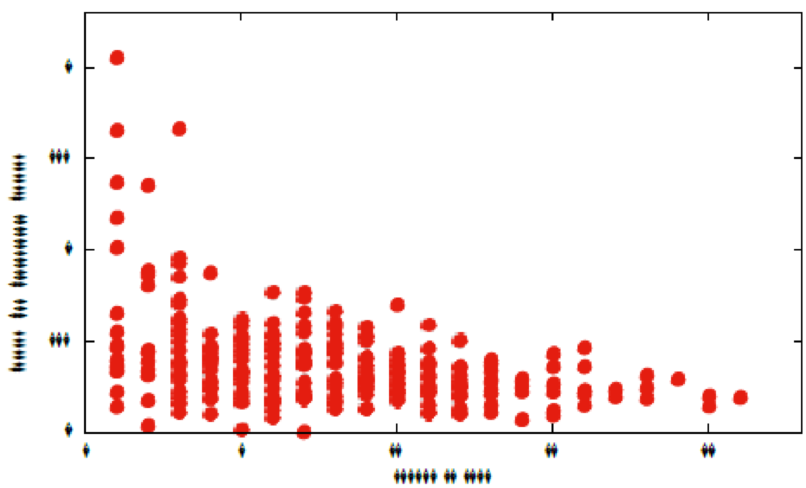


Figure 22: Latency inflation over shortest-path c-latency vs. number of hops in the shortest-path between each city pair

6.2 Comparison of latencies in AT&T's Network and measurements using a large CDN

In section 5 we described the measurements we performed between the servers of a large CDN. The CDN uses AT&T for network connectivity at multiple places and some of our measurements were between CDN servers both of which using AT&T as the network provider. 16 city pairs in AT&T's latency data have a fiber conduit between them marked as used/owned by AT&T in the InterTubes dataset(the left-most points in figure 22). Between 8 of these cities we have measurements in the CDN data, with both endpoints getting network connectivity from AT&T. For these city pairs, the comparison of the latencies published by AT&T and performed by us between the CDN servers is presented in table 3. In general, latencies in the CDN data are slightly higher than observed in AT&T data, except for Indianapolis and St. Louis pair.

The slightly larger latencies in the CDN data can potentially be explained by two factors: 1- Our measurements between CDN servers were performed in $3-6$ hour intervals over a few days, with 10 measurements on average per city pair, whereas the AT&T data covers roughly a 5-week period with data updated by AT&T every 15 minutes; 2 - Each CDN server is selected using a 25km radius from the conduit endpoint locations, resulting in a few extra hops (and distance) before routers in the core. Regardless, latencies observed in both networks are substantially higher than *c*-latency, except for Dallas Houston pair where minimum latency in AT&T data is only slightly higher (14%) than *c*-latency. For example, both latencies between Atlanta and Nashville are more than $2\times$ inflated in table 3 and the AT&T's published latency between Indianapolis and St. Louis is almost $3\times$ inflated over *c*-latency.

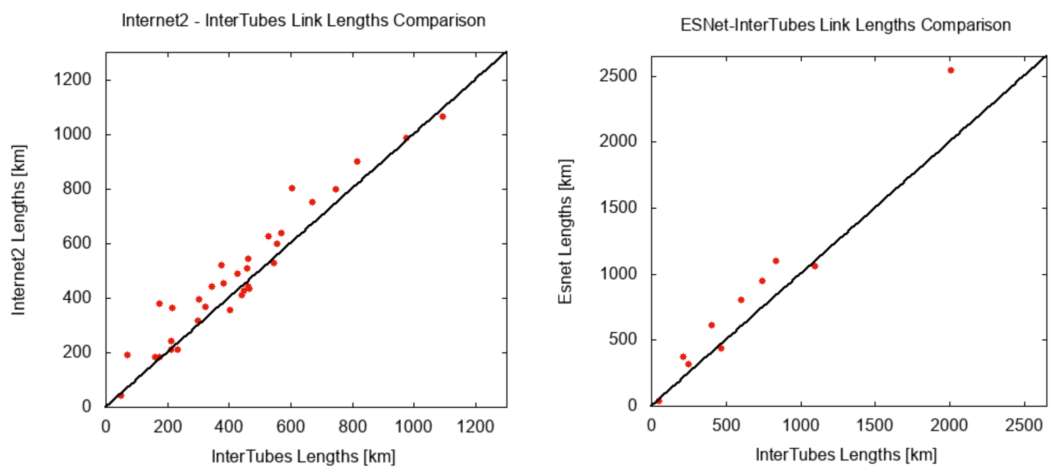
We shared this data with AT&T engineers to get their opinion about what might cause these large inflations. We received a comment about some of the link lengths in our data being somewhat short. We also learned that "depending on the transmission technology in use, dispersion compensation can increase the link latency by up to 17%". This happens because over large distances a single wave in fiber is dispersed into multiple colors traveling in different speeds, and extra spools of fiber are used at optical amplifiers to merge the signals traveling at different speeds. For example, a dispersion compensating fiber(DCF) system of length 1000 km would require 12 amplifiers, each containing a small spool of DCF. Assuming that standard single-mode fiber is used, then the total length of the DCF would be about 170 km, so each of the optical amplifiers would have about 14 km of DCF. Note that these are ball-park numbers, obtained through personal communication with an AT&T engineer. Mentioned 17% percent overhead due to DCF is consistent with the $15-25\%$ extra latency overhead due to DCF given in [4]. Even though DCF(extra fiber spools) might be a contributing factor, $2-3\times$ inflation in the observed latencies over some links still requires an explanation. The most logical explanation would be the use of a significantly longer fiber route, which might or might not be captured in the InterTubes dataset; however this is not something we are able to verify yet.

7 Comparison with other Research Networks

Since we obtained somewhat unexpected results with InterTubes dataset, we also examined link lengths and measured latencies in two research networks: Internet2 and ESnet. We obtained the long-haul fiber link lengths from both networks. Moreover, both networks constantly measure the throughput, loss and latency between their links and some of this data is available for researchers [11, 16]. First, we describe how link lengths in Internet2 and ESnet compare to lengths in InterTubes dataset. Then we describe the results of latency inflation analysis in ESnet.

Table 3: Comparison of latencies of the CDN and AT&T over fiber conduits in AT&T's backbone. Distances in kilometers, times in milliseconds.

City 1	City 2	Conduit Length(km)	CDN Min. RTT	AT&T Min. RTT	*c*-latency
Atlanta	Dallas	1418.38	18.32	17	13.91
Atlanta	Nashville	405.31	9.26	8	3.97
Dallas	Houston	446.69	6.55	5	4.38
Dallas	New Orleans	826.97	13.40	12	8.11
Houston	San Antonio	379.80	5.28	5	3.72
Indianapolis	St. Louis	434.21	7.35	13	4.26
Kansas City	St. Louis	458.10	6.28	6	4.49
Nashville	St. Louis	460.99	12.85	12	4.52



(a) *Lengths in Internet2 and InterTubes* (b) *Lengths in ESnet and InterTubes*

Figure 23: Comparison of lengths between common city pairs in the InterTubes dataset to ESnet and Internet2 backbones

7.1 Comparison of Link Lengths to InterTubes Dataset

Internet2 is the largest research and education network in USA. ESnet(Energy Sciences Network) is a high speed network owned by Department of Energy which not only connects the national labs in US among each other but also connects them to other research institutions in Europe. Internet2 and ESnet share a large portion of their network (88 long-haul links with 8.8 Tbps capacity) since the last major upgrade of these networks, done in partnership with Level3 Communications [7,20]. We obtained fiber link lengths from both networks[7]. Then we compared these to the lengths in the InterTubes dataset for all common links. Figure 23 shows the results of this comparison in a scatter plot for both networks. Even though there are points below the diagonal in each figure (links for which length in InterTubes dataset is larger), the difference in length for these links is mostly very small. On the other hand, for both networks, both the number and the magnitude of the difference of lengths are larger for the links in which the length is smaller in the InterTubes dataset[8].

Table 4: Latency Inflation in AT&T network with updated link lengths from Internet2 and ESnet. Distances in kilometers, times in milliseconds.

City 1	City 2	AT&T Latency	InterTubes Len.	Alt. Len.	*c*-latency	Inflation
Atlanta	Nashville	8.00	405.31	614.33	6.02	1.33
Chicago	Detroit	7.00	460.27	544.80	5.34	1.31
Chicago	Seattle	46.00	3414.75*	4062.00	39.82	1.16
Cleveland	Detroit	4.00	172.10	380.30	3.73	1.07
Dallas	Houston	5.00	446.69	426.00	4.18	1.20
Denver	Kansas City	15.00	1091.97*	1066.70	10.46	1.43
Houston	San Antonio	5.00	379.80	454.00	4.45	1.12
Kansas City	St. Louis	6.00	458.10	508.20	4.98	1.20
Philadelphia	Washington	4.00	246.66	377.60	3.70	1.08
Phoenix	San Diego	8.00	555.74	599.80	5.88	1.36

We reexamined the latency inflation in AT&T data over the links with a different length in Internet2 or ESnet compared to the InterTubes dataset. We were able to identify 10 such links (out of 16). Table 4 shows these links, their lengths, minimum latency in AT&T over them and inflation with the alternative link lengths obtained from Internet2 or ESnet data. All links except the one between Chicago and Seattle are used by AT&T and Level3 according to InterTubes dataset, and since Level3 is partnering with Internet2 it is plausible that the alternative lengths are true lengths. Note that two link lengths are marked with a $*$ in the table. For these two links, the published values in the InterTubes dataset are line-of-sight distances, and we inflated the length of all such links by 20%(the median link-length / line-of-sight distance ratio) before computing *c*-latency. The values in the table are inflated values. With the alternative link lengths, the median(maximum) inflation of AT&T latency to *c*-latency is found as 1.20(1.43). These are much smaller than in figure 21b, where the median and maximum inflation are 1.47 and 3.05. Considering some of this inflation can be accounted by dispersion compensation overhead, these inflation values seem to be closer to the truth.

[7]The source of the link lengths we obtained from ESnet are latency tests in the optical layer. Using 204, 000 km/s as the speed of light in fiber to estimate latency over the obtained link lengths, our latency estimate is on average within 45 μs of the latencies obtained in the physical layer measurements performed at ESnet. We are trying to verify whether the link lengths in Internet2 are obtained from physical layer latency tests as well

[8]There is one additional data point not shown in figure 23a. For this link the line-of-sight distance is provided as the link length in the InterTubes dataset. The link length in Internet2 data is 1215 km longer than line-of-sight distance.

7.2 Latency in ESnet network

We obtained latency measurements from both Internet2 and ESnet. From Internet2 measurement archive, we obtained latency measurements between all pairs of 11 core router locations for one day (April 12, 2017). The measurements are performed with One Way Active Measurement Protocol(OWAMP) [27] every 10 ms, and we obtained aggregated statistics for every minute of the day. Unfortunately for an overwhelming majority of the links, we observed that the one-way delay is smaller than the lower-bound dictated by the speed-of-light in fiber. The clocks on both ends are synchronized using Network Time Protocol(NTP), and the clock skew was significant enough for our purposes, preventing evaluating the latency along fiber links of a few hundred km long. So, we focus on measurement data that we obtained from ESnet.

Table 5: Latency inflation in ESnet backbone over directly connected POPs

	Min.	**Median**	**Max.**
Min. RTT	1.010	1.031	1.215
Avg. RTT	1.014	1.048	1.250

We downloaded traceroutes performed between all directly connected major POPs in ESnet backbone [12]; in total we have traceroutes in both forward and reverse directions for 24 long-haul links. Our data covers a 3.5 month period between January 1 and April 18, 2017. Traceroutes were performed every 10 minutes. On average we have 13.8K traceroutes for each link and the minimum number of traceroutes for a link is 8.6K. Each hop on the traceroutes has names identifying its location, so we were able to verify that the packets followed the target fiber link in each traceroute. Results are summarized in table 5, where we show the inflation over all the links both for minimum and average RTT observed for each link during the measurement period. We see that even the average RTTs are much closer to the speed-of-light lower bound compared to minimum RTTs in the CDN and AT&T data. However this is not very surprising, since we were able to verify that the paths in traceroutes followed the fiber conduits and we have highly accurate fiber lengths for ESnet as explained previously (in section 7.1). The largest inflation is observed for the shortest link we examined (between Sacramento and Sunnyvale, 244 km), and the inflation in minimum latency in one direction was 22%. However the absolute value of the difference between minimum latency and c-latency is only 0.3 ms in one direction and 0.5 ms in the other, which is small.

Previously a small scale study of latency inflation in optical networks is performed in [23]. The authors had accurate link lengths and detailed knowledge of the existing network elements along the links they measured including Dispersion Compensating Modules (DCM) and their characteristics. They calculated RTTs based on this information and compared it to RTTs observed with ping. Along the same lines with our conversations with AT&T engineers, dispersion compensation added a 15% latency overhead in their calculations. Despite the detailed knowledge of network, measured and calculated RTTs differed by 9% for some links. The authors mention that this difference is probably due to congestion, but they did not provide any information about the duration and frequency of their measurements.

8 Conclusion

The measurements we performed between CDN clusters and the latency data we obtained from a large ISP(AT&T) exhibit much larger inflations over the speed-of-light along the fiber lengths than we anticipated. Here we discuss a few potential contributors of this outcome.

One obvious factor is the uncertainty we have in terms of the reliability of the InterTubes dataset we used. We have uncertainty not only in the endpoint locations of the fiber conduits but also in the provided link lengths. Despite the authors doing their best in collecting and verifying the link lengths, it seems the quality of the link lengths is probably not uniform. Some of these lengths are obtained from the ISPs owning the links and these are probably more accurate than the rest. For others, the authors calculated lengths based on the route it takes over the surface of the Earth by computing line-of-sight distances for each segment in the fiber route. The routes seem to be obtained by visually inspecting published maps, railroads and other right-of-ways which fiber conduits typically follow. It is possible that the link lengths that are shorter than their counterparts in ESnet or Internet2, and the lengths that are described as being ”somewhat short” by the AT&T engineers are obtained this way. However, it is very difficult to be certain about this without examining the routes themselves as there are often multiple fiber routes between same pairs of locations with markedly different lengths [26].

Another factor is the ”hidden” sources of latency in the optical layer even in the case of accurate lengths based on fiber routes. Based on the transmission technology used inside the fiber, there might be additional spools due to dispersion compensation, increasing the distance the light needs to travel inside the fiber. Unfortunately the fibers owned/used by different ISPs are laid out in different times, and depending on the available transmission technology and the cost at the time, they may have very different characteristics. The only way to be certain about the actual lengths is by computing them based on tests in the physical layer. However, this is difficult to obtain in most cases. It is possible that some of the lengths in the InterTubes dataset (the ones obtained from published ISP data) are based on tests in the physical layer, but it is not possible to identify which.

Yet another factor is our use of traceroutes to obtain RTTs along the fiber conduits, with no visibility below IP layer. We did geolocate the routers in the traceroutes and try to understand which measurements followed routes along the intended conduit. Despite errors in geolocation, we were able to find measurements which showed with high likelihood the path along the fiber conduit was taken. However, for some of these measurements the RTTs are still higher than we expect and we cannot determine whether this is due to inaccurate link lengths or longer MPLS tunnels below the IP layer.

It is encouraging to note that the RTTs we observed in ESnet are much closer to the physical limit imposed by speed-of-light. However, for these measurements we not only had very accurate link lengths obtained from tests in the physical layer, but also were able to verify that traceroutes followed the intended fiber conduits.

9 References

[1] Akamai Technologies, Inc. EdgePlatform. http://www4.akamai.com/html/technology/products/edgescape.html, 2013.

[2] M. J. Arif, S. Karunasekera, and S. Kulkarni. GeoWeight: Internet Host Geolocation Based on a Probability Model for Latency Measurements. In *Proceedings of the Thirty-Third Aus- tralasian Conferenc on Computer Science - Volume 102*, ACSC '10, pages 89–98, Darlinghurst, Australia, Australia, January 2010.

[3] ATT. U.S. Network Latency. https://ipnetwork.bgtmo.ip.att.net/pws/network_delay. html

[4] Vjaceslavs Bobrovs, Sandis Spolitis, and Girts Ivanovs. Latency causes and reduction in optical metro networks. In *Proceedings of SPIE*, volume 9008, 2013.

[5] Ilker Nadi Bozkurt, Anthony Aguirre, Balakrishnan Chandrasekaran, Philip Brighten Godfrey, Gregory Laughlin, Bruce MacDowell Maggs, and Ankit Singla. Why is the internet so slow!? In *ACM Passive and Active Measurement Conference 2017 (PAM 2017)*, 2017.

[6] Nicole Caruso. A Distributed System For Large-Scale Geolocalization Of Internet Hosts. diploma thesis, Cornell University, Ithaca, NY, 2011.

[7] Doug Howell. Nation's First 100G Open, Nationwide, Software-Defined Network Launches for Education, Research, Industry and Innovators. https://www.internet2.edu/news/detail/ 2506/.

[8] Ramakrishnan Durairajan, Paul Barford, Joel Sommers, and Walter Willinger. Intertubes: A study of the us long-haul fiber-optic infrastructure. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 565–578, New York, NY, USA, 2015. ACM.

[9] Brian Eriksson, Paul Barford, Bruce Maggs, and Robert Nowak. Posit: a lightweight approach for IP geolocation. *SIGMETRICS Perform. Eval. Rev.*, 40(2):2–11, October 2012.

[10] Brian Eriksson, Paul Barford, Joel Sommers, and Robert Nowak. A Learning-Based Approach for IP Geolocation. In *Proc. of the 11th International Conf. on Passive and Active Measure- ment*, PAM'10, pages 171–180, Berlin, Heidelberg, April 2010.

[11] ESnet. Data for Researchers. https://www.es.net/network-r-and-d/ data-for-researchers. Last accessed: July 03, 2017.

[12] ESnet. ESnet 100Gbps Routed Network Map. https://www.es.net/assets/ 20160719-ESnetBackbone.pdf. Last accessed: July 3,2017.

[13] Phillipa Gill, Yashar Ganjali, Bernard Wong, and David Lie. Dude, where's that IP? Circumventing measurement-based IP geolocation. In *Proceedings of the 19th USENIX conference on Security*, USENIX Security'10, pages 16–16, Berkeley, CA, USA, 2010. USENIX Association.

[14] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. Constraint-Based Geolocation of Internet Hosts. In *ACM Internet Measurement Conference*, Taormina, Sicily, Italy, October 2004.

[15] Chuanxiong Guo, Yunxin Liu, Wenchao Shen, H.J. Wang, Qing Yu, and Yongguang Zhang. Mining the web and the internet for accurate ip address geolocations. In *INFOCOM 2009, IEEE*, pages 2841–2845, 2009.

[16] Internet2 Network NOC. Bandwidth and Latency Status. http://noc.net.internet2.edu/ i2network/live-network-status/internet2-network-bandwidth-and-latency-status. html. Last accessed: July 03, 2017.

[17] IP2Location.com. IPInfoDB. http://www.ip2location.com/, 2013.

[18] Ethan Katz-Bassett, John P. John, Arvind Krishnamurthy, David Wetherall, Thomas Ander- son, and Yatin Chawathe. Towards IP Geolocation Using Delay and Topology Measurements. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, pages 71–84, New York, NY, USA, October 2006.

[19] S. Laki, P. Mátray, P. Hága, T. Sebok, I. Csabai, and G. Vattay. Spotter: A Model Based Active Geolocation Service. In *IEEE INFOCOM*, April 2011.

[20] Level3 Communications. Internet2 and Level 3 to Provide Unprecedented Amount of Network Capacity to Research and Educational Institutions in Support of Telemedicine, Distance Learning and a Range of Research. goo.gl/kDg1Qm.

[21] MaxMind, Inc. GeoIP City. http://www.maxmind.com/en/geolocation_landing, 2013.

[22] Net Industries, LLC. hostip.info. http://www4.akamai.com/html/technology/products/ edgescape.html, 2013.

[23] Peter Noutsios and Vipin Patel. Optical latency on a core network. *Journal of Optical Networking*, 3(4):242–246, Apr 2004.

[24] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Proceedings of ACM SIGCOMM confer- ence*, San Diego, CA, USA, August 2001.

[25] Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. IP Geolocation Databases: Unreliable? *SIGCOMM Comput. Commun. Rev.*, 41(2):53–56, April 2011.

[26] Riot Games, Engineering Department. Fixing the Internet for Real Time Applications. http://engineering.riotgames.com/news/fixing-internet-real-time-applications-part-ii.

[27] S. Shalunov, B. Teitelbaum, A. Karp, J. Boote, and M. Zekauskas. RFC 4656 - A One-way Active Measurement Protocol (OWAMP).

[28] Yong Wang, Daniel Burgener, Marcel Flores, Aleksandar Kuzmanovic, and Cheng Huang. Towards Street-Level Client-Independent IP Geolocation. In *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, NSDI'11, pages 27–27, Berkeley, CA, USA, April 2011.

[29] Bernard Wong, Ivan Stoyanov, and Emin Gün Sirer. Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts. In *NSDI*, April 2007.

[30] Inja Youn, Brian L. Mark, and Dana Richards. Statistical Geolocation of Internet Hosts. In *ICCCN*, pages 1–6, 2009.

List of Acronyms

CBG	Constraint-Based Geolocation
CDF	Cumulative Distribution Function
CDN	Content Delivery Network
DCM	Dispersion Compensating Modules
DNS	Domain Name Service
ECDF	Empirical Cumulative Distribution Function
GPS	Global Positioning System
IP	Internet Protocol
ISP	Internet Service Provider
km	kilometers
OWAMP	One Way Active Measurement Protocol
RTT	Round Trip Time
TCP	Transmission Control Protocol
TPG	Topology Based Geolocation